1992-03

The development of a scheduling application in support of the a paperless ship.

Hale, Richard J.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE DEVELOPMENT OF A SCHEDULING APPLICATION

IN SUPPORT OF THE PAPERLESS SHIP

by

Richard J. Hale

March 1992

Thesis Advisor: C. Thomas Wu

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (*If applicable*) 37	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11 TITLE (*Include Security Classification*)
THE DEVELOPMENT OF A SCHEDULING APPLICATION IN SUPPORT OF THE PAPERLESS SHIP

12 PERSONAL AUTHOR(S) Richard John Hale

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*year, month, day*)	15 PAGE COUNT
Master's Thesis	From To	March 1992	64

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	

19 ABSTRACT (*continue on reverse if necessary and identify by block number*)

The scheduling of events aboard U.S. Navy ships is a complex and dynamic problem. Currently, this process is primarily manual and involves searching through several manuals and instructions to find information. Many times, the schedules produced are inaccurate, which makes conducting activities very difficult and results in crew frustration. By automating some of the functions of the scheduling process, accurate schedules can be quickly produced. As a result, valuable time will be saved and the planning and coordination of shipboard activities can be effectively ccomplished in order to achieve and maintain a high level of readiness. This thesis is part of the ongoing Argos research project which supports the Navy's paperless ship concept by eliminating or minimizing manual procedures used on ships.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL C.T.Wu	22b TELEPHONE (*Include Area code*) 408 646 3391	22c OFFICE SYMBOL 52Wq

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

Approved for public release; distribution is unlimited.

THE DEVELOPMENT OF A SCHEDULING APPLICATION IN SUPPORT OF THE PAPERLESS SHIP

by

Richard J. Hale
Lieutenant, United States Navy
B.S., United States Naval Academy, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

ABSTRACT

The scheduling of events aboard U.S. Navy ships is a complex and dynamic problem. Currently, this process is primarily manual and involves searching through several manuals and instructions to find information. Many times the schedules produced are inaccurate, which can make conducting activities very difficult and result in crew frustration. By automating some of the functions of the scheduling process, accurate schedules can be quickly produced. As a result, valuable time will be saved and the planning and coordination of shipboard activities can be effectively accomplished in order to achieve and maintain a high level of readiness. This thesis is part of the ongoing Argos research project which supports the Navy's paperless ship concept by eliminating or minimizing manual procedures used on ships.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The scheduling of events in any organization is a dynamic process which can require a manager's constant attention. Aboard U.S. Navy ships, the Operations Officer schedules and coordinates events such as underway periods, inspections and exercises. The command's schedule is developed following the guidance of superiors in the chain of command. This process is complicated and changes usually occur with very short notice at all levels in the chain of command.

The process at a shipboard level is normally manual. That is, most schedules are written on paper or cardboard. As changes occur, these schedules are manually corrected. This process is extremely slow, tedious and can lead to serious errors. Since this process is manual, many schedulers delay making the necessary updates or fail to make them at all. In the end, the ship's crew suffers because they are not kept properly informed of changes, which can lead to frustration and low morale. One improvement to the scheduling process could be to automate. For instance, by eliminating the need to search through manuals for current information, a scheduler could be more efficient and effective.

Argos is an ongoing research project at the Naval Postgraduate School. Its goal is to automate many of the manual activities on U.S. Navy ships and eventually make them less dependent on paper. All of the Argos applications feature a friendly graphical user interface which supports text and, to some extent, sound. Thus, anyone can quickly become proficient with Argos, whether a computer beginner or expert.

The Argos Scheduling Module was developed using the Hypercard scripting language which is now available as part of Macintosh system software.[1] Some work has been done to develop an application on personal computers using Microsoft Diskette Operating System (MS-DOS); however, most of the applications developed have been on Macintosh computers.[2] The need for an automated scheduling aid is great. Factors which can control a ship's schedule are numerous and changes occur constantly. The purpose of this thesis is to demonstrate the feasibility of a scheduling tool that can be used to automate and simplify the manual procedures which are in use today.

---

[1]Macintosh and Hypercard are registered trademarks of Apple Computer, Inc.

[2]Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

This thesis is organized as follows: Chapter II discusses the scheduling process as it currently exists. Hardware constraints are presented in Chapter III. Chapter IV discusses the design and use of the scheduling prototype. Chapter V discusses the benefits of automating the scheduling process and conclusions and recommendations for follow on thesis work are presented in Chapter VI. Hypercard scripts of the major functions of the prototype are contained in the appendix.

II. BACKGROUND

An accurate schedule of events is critical to any organization. For U.S. Navy ships a schedule is the foundation upon which all other activities build. For example, maintenance personnel must have up to date knowledge of underway periods to facilitate the scheduling and accomplishing of preventive maintenance on major pieces of machinery which will affect main propulsion. Some activities require assistance from other commands or units, such as a crane which is used to remove a radar or other antenna from the ship's mast. Additionally, personnel must know when underway periods and inspections are scheduled in order to plan personal items, such as extended travel requiring leave.

A ship's Supply Officer determines how much money will be needed to support any special activities based upon upcoming events such as port visits. A ship's expenses depend upon whether the ship will moor to a pier or whether it will anchor out in a harbor as determined in the schedule. Perhaps one of the most important aspects of knowing a ship's schedule involves personnel training. Although simulators and trainers are available for many different watch stations, most personnel qualifications on a

ship require performing specific activities while the ship is underway. Several qualifications must be completed within time limits. Surface Warfare officer qualifications must be obtained within a reasonable period of time aboard ship or one could be dismissed from the Navy. If personnel can look at a list of events and determine when underway times are scheduled, then training times can be allocated to ensure all qualifications are achieved. This is especially critical when several people are competing for the same qualification. Finally, there are teams which must be trained. For example, several fire fighting teams must be qualified. The teams are comprised of members from every department on board a ship and training requires the personnel to be off the ship for two to three days. This evolution must be scheduled and promulgated well in advance so the impact of personnel shortages can be properly handled.

In spite of its importance, the scheduling process as it exists on most U.S. Navy ships today is prone to error and confusion. Each individual ship develops a schedule following a general outline issued from its squadron. Each squadron coordinates the activities of ten to fifteen ships. The squadron coordinates its scheduling up through the chain of command. Generally, about once every three months a ship will receive new guidelines from its squadron listing events which must be accomplished. Major underway periods are

already scheduled in these guidelines but other events can be scheduled by the ship. This task is most often performed by the Operations Officer, usually a senior lieutenant. Commonly, the process is completely manual and consists of listing events on a big piece of paper or cardboard. As changes occur, new schedules are made or old schedules are erased and annotated to reflect current information. Copies of these schedules are then made and distributed throughout the ship. Copies may be of poor quality either because of poor handwriting or faulty copiers. The end result is that many people cannot read or understand these schedules which leads to frustration and dissatisfaction.

The manual process described above has many inherent disadvantages including the following:

- the scheduler must search several instructions and manuals to determine event periodicity and event prerequisites.

- if old schedules are lost or destroyed, the formulation of new schedules is extremely error prone.

- because the process is manual, the extra instructions and old schedules add weight to a ship which negatively affects stability and fuel efficiency.

- the time consuming nature of these tasks is a deterrent to making changes in a timely manner. Sometimes it is impossible to keep up with changes as they occur. Such is often the case when a ship is underway and schedules can change every hour.

- since the process is so confusing usually one person performs all the necessary functions. Again, due to

routine personnel turnover, a new scheduler must familiarize himself with all of the procedures and manuals.

One way to improve the scheduling process is to automate activities which are now done by hand. Of particular importance is the elimination of searching various sources for information about each event. The application must be an improvement over current methods and must be designed with a ship's space and weight limitations in mind. Finally, the development and implementation of this application must be cost effective. [Ref. 1:pp. 5-6]

Throughout the development of the Argos Scheduling Module, these considerations were used as guidelines. Ultimately, with the help of this application, the Operations Officer could have an assistant perform some of the scheduling functions and conduct periodic checks to ensure completeness and accuracy. This would allow more than one person to have detailed knowledge about the process which would alleviate some of the problems that are caused by personnel turnover.

III. OPERATIONAL REQUIREMENTS

The purpose of this thesis is to develop a prototype for automation of a ship's scheduling process. The Argos Scheduling Module can enhance the performance of a ship. The purpose of this chapter is to define the hardware features necessary to make this system easy to learn and use.

The computer must be a platform that can support graphics and sound and must be small enough to fit aboard a ship with limited space [Ref. 1:p. 7]. It must also be rugged enough to withstand the rocking and vibration common to a shipboard environment. The speed at which this computer must perform is critical for some of the Argos applications. Finally, price is always a factor when considering any new system. Because of the advances made in the computer industry, a powerful computer system can be purchased for a very reasonable price.

There are several choices of computer hardware that can be selected to support the Argos Scheduling Module. Most of the applications for the Argos project to date have been developed on Macintosh computers utilizing the Hypercard programming language. There are, however, existing software packages that can be used to develop similar applications on

IBM compatible computers which use Microsoft Disk Operating System (MS-DOS).

All Macintosh computers are able to support sound and are capable of being connected to a network if the appropriate software is installed. Any Macintosh computer that uses System 7 operating system automatically has the appropriate networking software installed. For Macintoshes without System 7 operating system, the networking software must be purchased separately. Usually the ability to support sound and to connect to a network are not part of the initial purchase of an IBM compatible computer. These capabilities require the purchase of additional computer cards which raises the final price of a system. These are important factors which must be considered. Although the sound and network features will make the initial purchase price higher, it is cheaper to buy them installed than it is to buy them later. [Ref. 1:pp. 7-8]

Since the acceptance and success of an application largely depends upon its user interface, an adequate monitor for the computer system is an important consideration. Monitors come in various shapes and sizes and can either be black and white or color. Over the last few years, monitors have become quite inexpensive. High resolution color monitors can be purchased for approximately $500. A complete Macintosh computer system with a color monitor

operating at speeds ranging from 16MHZ to 33MHZ can be purchased for approximately $3000. [Ref. 1:pp. 7-9]

The Hypercard language is a logical software tool because it supports prototyping. Prototyping can be defined as a working application that can be created quickly [Ref. 2]. The purpose of the prototype is to allow users to evaluate a computer application or system prior to its final implementation. Thus the prototype development should be inexpensive and flexible. Because Hypercard fulfills all of these requirements it was chosen for this and previous Argos projects.

The development of an application in Hypercard requires the use of one or more stacks. A Hypercard stack is a homogeneous collection of information. Each stack consists of several cards. A card can contain text or graphics and a new card can be added as more information is accumulated. A text field is used to write text on a card. Finally, buttons are used in cards and stacks to allow the user to operate. For example, a user can click his mouse on a button and go to a different card or to a different stack. Another use of a button could be to play sound or musical tunes. The possibilities for a button in any application are numerous. [Ref. 3:pp. 27-35]

In the Argos Scheduling Module, there are six stacks. When users open a stack for the first time, they will notice that the size of every card in the stack is nine inches.

This was designed to allow the application to be run on the earlier versions of Macintosh computers which have smaller screens. On newer computers with larger monitors, the card may not fill the entire screen. The card size can be adjusted accordingly, depending upon screen size and the amount of RAM available to Hypercard [Ref. 3:p. 106]. It is best to leave the card size at nine inches so the application can be used on any size monitor. When the size of a card is adjusted, some of the buttons and graphics may appear slightly offset from their original position. In these instances each item must be repositioned separately.

The six stacks of the Argos Scheduling Module and their cards, fields, and buttons perform many of the functions necessary for the scheduling process. The information required for scheduling, such as event periodicities and prerequisites, are also stored in these stacks. Some of the information is stored in fields which are visible to the user. Other information, such as the amount of fuel burned per day for a particular ship's class is hidden from the user and only brought into view as necessary. Finally, there is information in fields that is completely hidden but accessed by an application. No matter what type of field is used, it is important to realize that these fields are a type of database for any application developed in Hypercard.

Hypercard is used for prototyping because it provides the capability to store and manipulate information on a

limited basis. It is not intended to be used in the final production system. Research into using a relational database in the Argos project is ongoing. In a fully operational system, a database management software (DBMS) would be used. A DBMS permits users to create, maintain and extract information from the database.

A major distinction between Hypercard and database management software exists and must be explained. Hypercard lets a user move to other stacks to view or retrieve information as needed. One benefit of DBMS is its ability to generate reports. DBMS can also perform various functions on the output or send it to a printer. Hypercard is not designed for generating reports but is optimized for quickly looking through existing cards in search of desired information. The links between cards or stacks that are established in Hypercard are not rigid or finite and can be changed as new information is added. [Ref. 3:pp. 8-9]

IV. SYSTEM PROTOTYPE

A. DESIGN

Four stacks representing different ship classes were developed. In each of these, there are five cards which are used to collect information about each ship within these classes. The following is a description of each of these cards within the four stacks.

The first card welcomes the user to the Argos Scheduling Module and is the only card of its type in this project (Figure 1). The second card introduces the user to the types of ships which can be scheduled (Figure 2). Each silhouette represents a ship's class that is handled by this application [Ref. 4:p. 33]. Each ship is actually a transparent button which will lead the user to a different stack, each designed for one or two classes of ships.

As in Figure 2, there are six buttons which appear on every card in the four ship stacks. Two of these buttons are arrows which are located at the top of each card. The arrow which points to the right is used to advance the user one card in that direction. The arrow which points to the left performs the same action in a backward direction. On the bottom right of every card are four buttons, all of which utilize sound [Ref. 4:p 33]. These buttons are called

Figure 1
Scheduling Welcome Screen

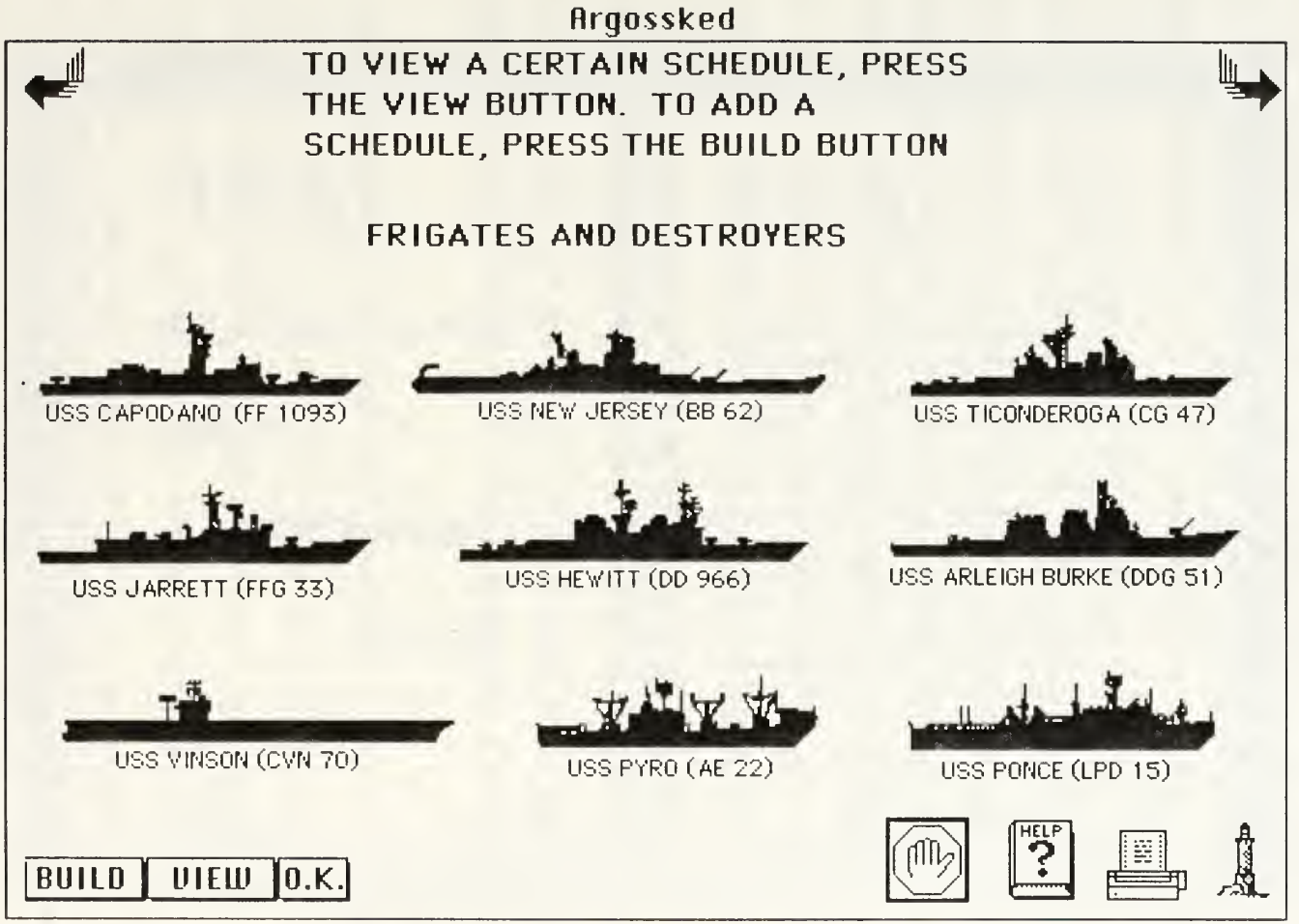

Figure 2
Ship Silhouettes

icons. An icon is a graphic image that can be used on screen displays [Ref. 5:p. 565]. Sound was installed in these buttons by using a stack called TALK TO ME! written by James L. Paul [Ref. 6]. The stack TALK TO ME! uses XCMDS which are compiled routines written in a language other than Hypercard which can do something that Hypercard cannot: in this case create sound [Ref. 7]. The first button, which looks like an open palm, is used to quit this application and Hypercard. The second button will enable a HELP stack, which provides a brief description of the design of this project and how to use the buttons. One card of this stack is shown in Figure 3. [Ref. 4:p. 134] The next button looks like a printer and is used to print out the card being viewed. The last button resembles a lighthouse and removes the user from this application but does not quit Hypercard. The sound alerts the user that these buttons perform special functions.

Moving to the next card the user will see six text fields called scrolling fields (Figure 4). By clicking the mouse on one of the arrows along the right edge of these fields additional information can be seen. In the first text field there is a list of inspections and events. Some of these are marked with an asterisk to signify there are special circumstances to be considered when scheduling them. By clicking the mouse on the EVENT button the user will see another field in which inspection prerequisites are

Figure 3
Help Stack Screen 1

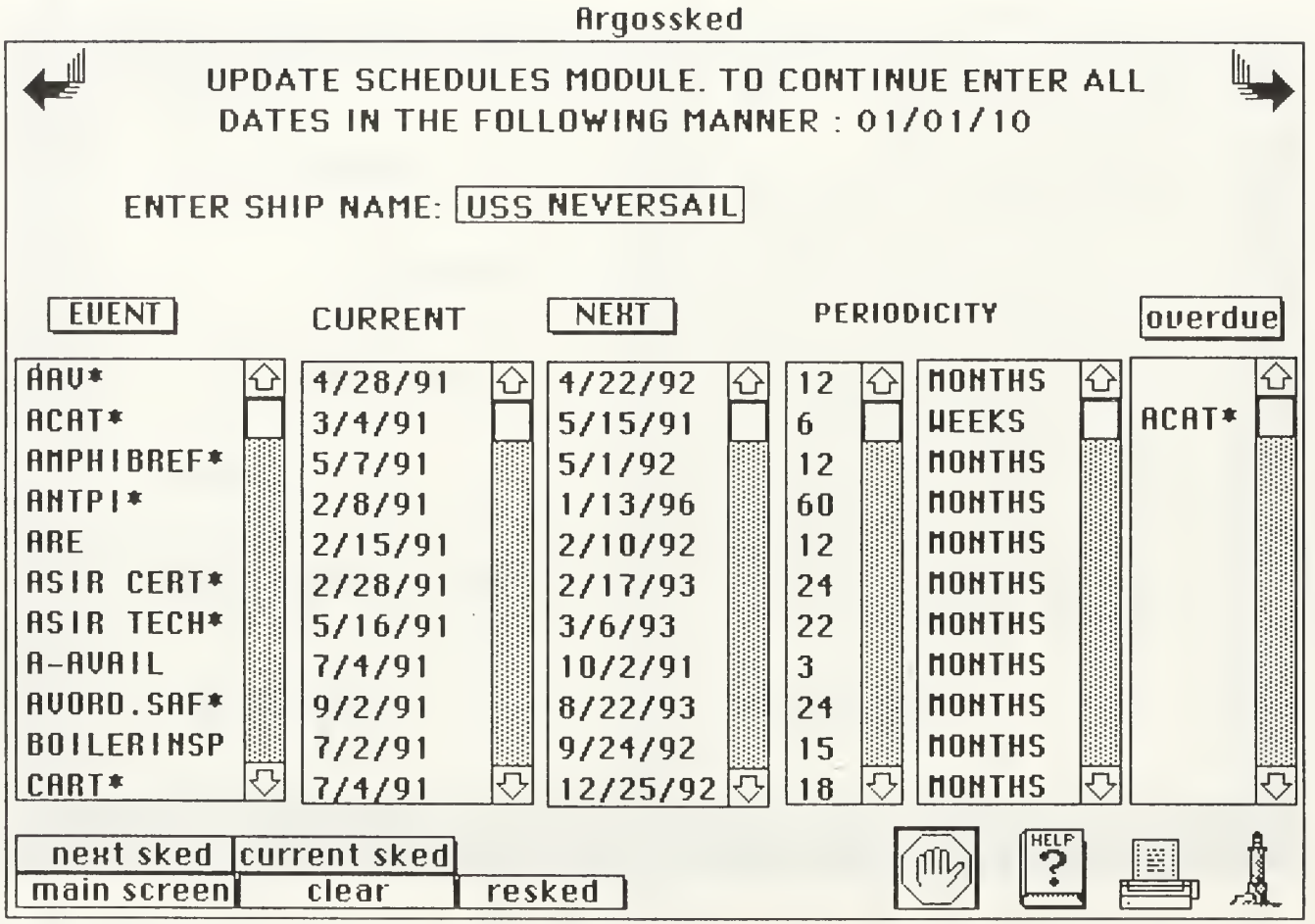

Figure 4
List of Inspections and Events

listed (Figure 5) [Ref. 8]. The asterisk ensures an inspection or event is properly scheduled. The second purpose of this card is to activate the computation of when each inspection or event can be scheduled. These dates are computed using the periodicity listed [Ref. 8]. Of course, the date that an inspection or event will actually be completed may vary but these computed dates provide guidelines for the scheduler. The third important function performed by this card is to compute any event not completed within the required periodicities. This information is vital because a ship must maintain a certain level of training and readiness at all times in order to complete upcoming operations. The text in the fields labeled EVENT and PERIODICITY cannot be erased or modified by certain users. These fields have a property called locked text [Ref. 3:p. 143]. This feature can only be activated or deactivated by someone with the proper authority and user level [Ref. 3:p. 25]. This was designed to ensure the integrity of the scheduling information.

The next card contains text fields which hold the current and next schedules, listed in chronological order (Figure 6). A third text field holds the events of a specific quarter of either the current or next schedules, listed in chronological order. These fields were designed to provide the scheduler the ability to view a list of

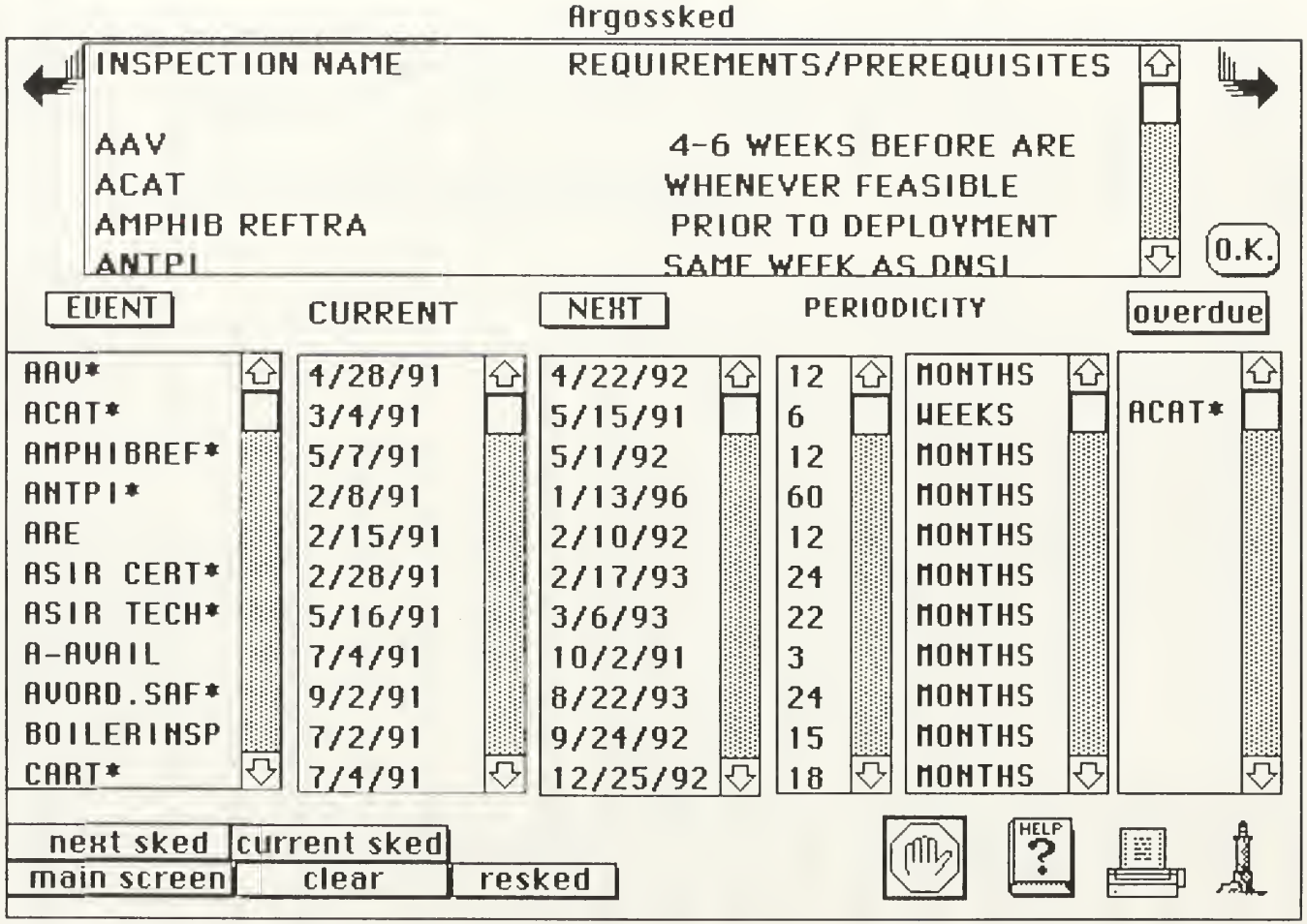

Figure 5
List of Inspections and Events with Prerequisites

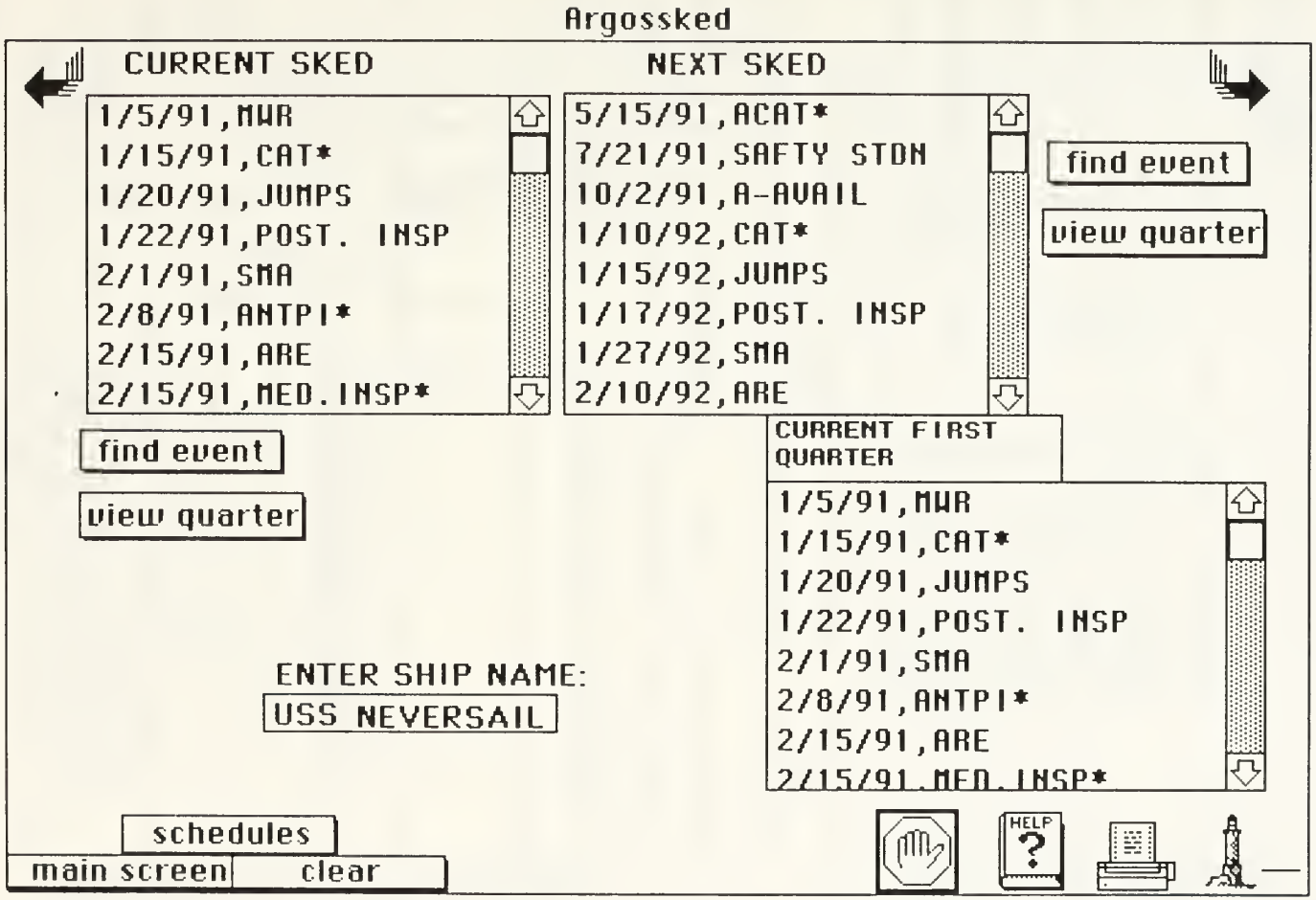

Figure 6
Current and Next Schedules

several events at a glance which facilitates long range planning.

The function of the next card is to compute fuel consumption (Figure 7). The DBSR button displays different classes of ships and their respective daily burn rate for fuel (Figure 8) [Ref. 9]. The first text field contains a list of employment terms and their respective fuel ratios. Both the field which holds daily burn rates and the field which holds employment terms are locked text. The purpose of this card is to provide a scheduler with a computation of the amount of fuel required for specific events. For example, the total amount required for a certain month or quarter can be computed based on upcoming events. The user is also provided with the number of barrels which can be used on a daily basis while underway. This feature allows either underway or inport refuelings to be scheduled in advance.

The final card that will be used for every ship is the Personnel Tempo Reporting Card (Figure 9). This card activates the computation of days personnel spend in homeport versus the number of days spent away from homeport. This is done each quarter and the quarterly point total is added to a cumulative point total in order to monitor trends. This trend is also displayed on this card. Finally, the scheduler can then enter the recovery date in the appropriate text field. The recovery date is defined as

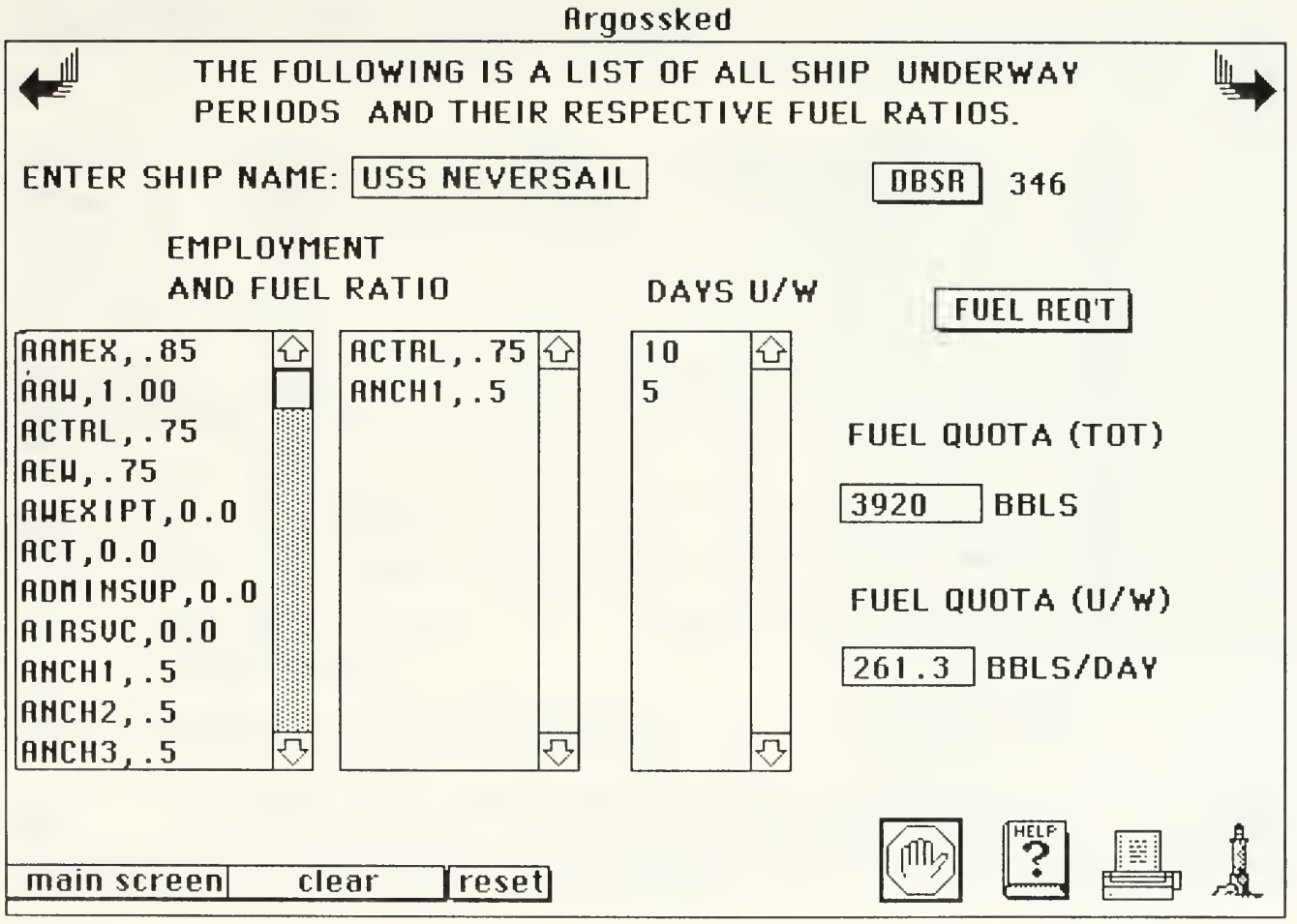

Figure 7
Fuel Quota Card

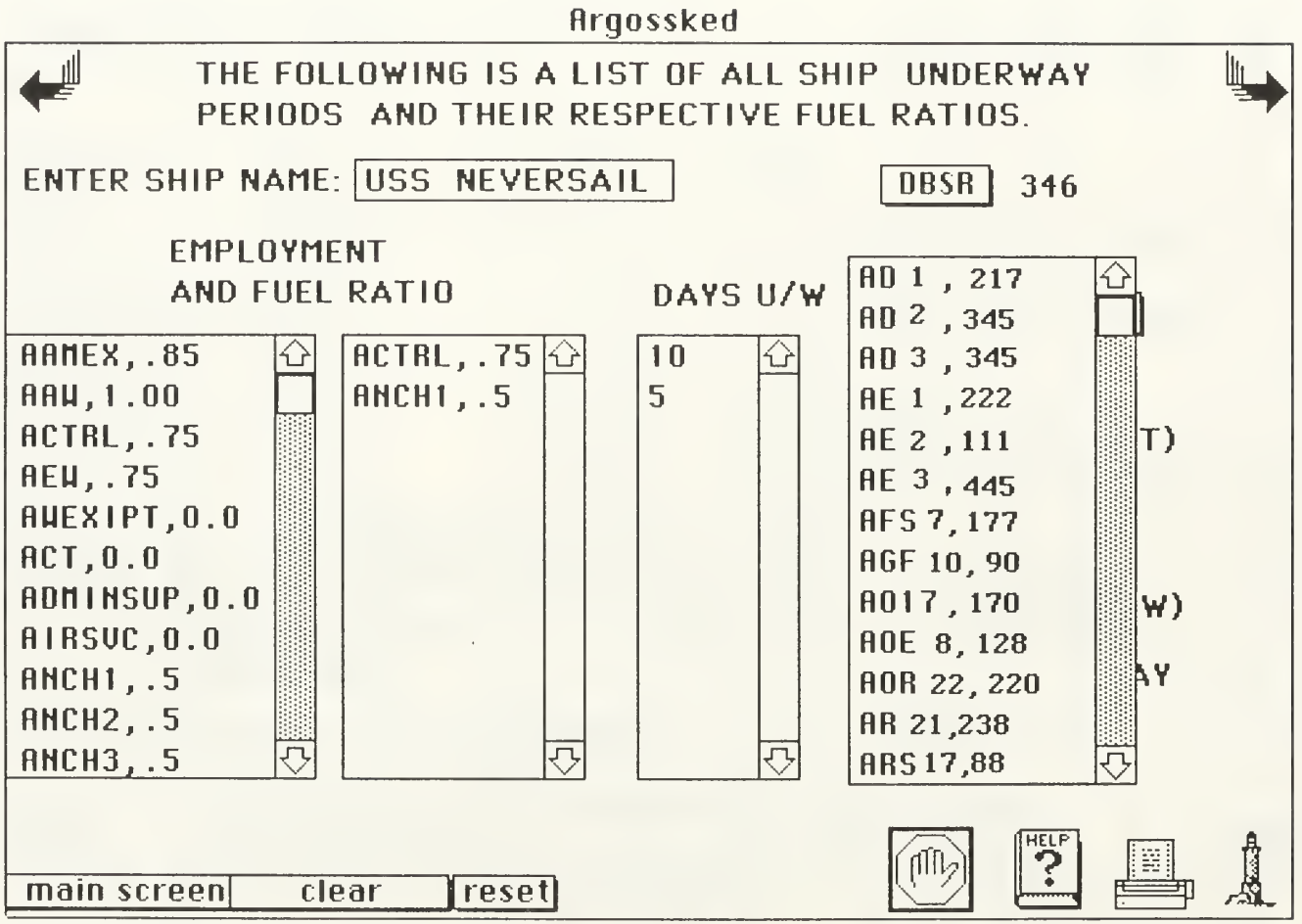

Figure 8
Fuel Quota Card with Fuel Burn Rates

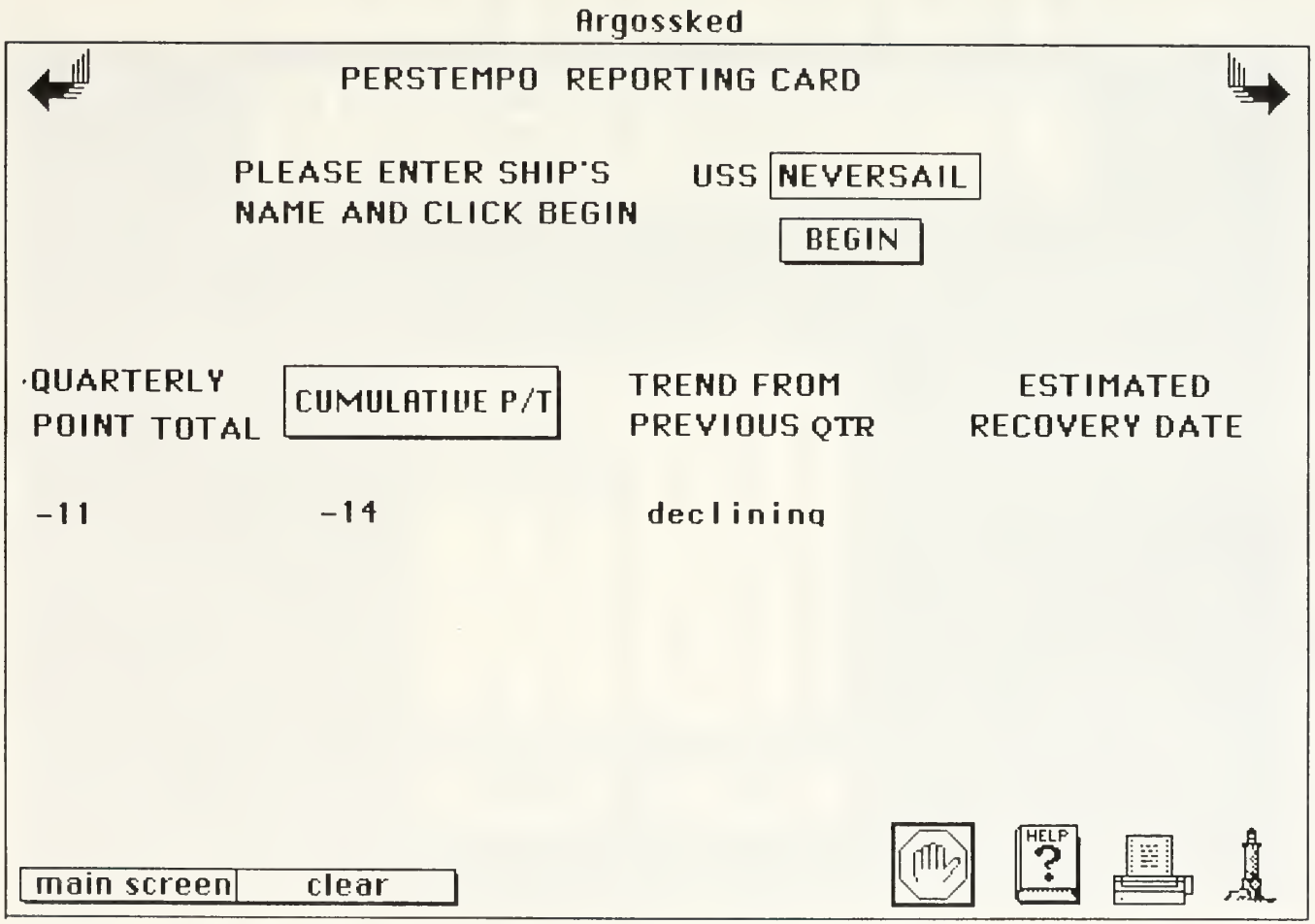

Figure 9
Personnel Tempo Reporting Card

the date when a unit will meet personnel tempo minimum standards. The minimum is defined by deployment length and turn around ratio (TAR). TAR is the ratio between the number of months a unit spends between deployments and the length of the last deployment. [Ref. 10]

The SCHEDULES stack is where the final schedule for a ship is printed out using a calendar (Figure 10 and Figure 11). The calendars are taken from a stack called STACK TEMPLATES which comes as a part of Hypercard. All stacks for the different types of ships were designed the same. Thus, users will see the same buttons throughout, which cuts down on the amount of time required to learn this application. Although the scheduling stack is different, schedulers should quickly learn how to use it effectively. The scheduling stack is really the end product; a useful schedule is produced. The other stacks act as worksheets. The scheduler inputs data in other stacks and that data is organized and displayed in the SCHEDULES stack so the user can quickly scan a schedule and base decisions on it. Thus the overall objective of making scheduling easier and more effective has been achieved.

B. DESCRIPTION

The easiest way to begin using the Argos Scheduling Module is to double click the mouse on the stack named

Figure 10
Schedules Stack Introduction Screen

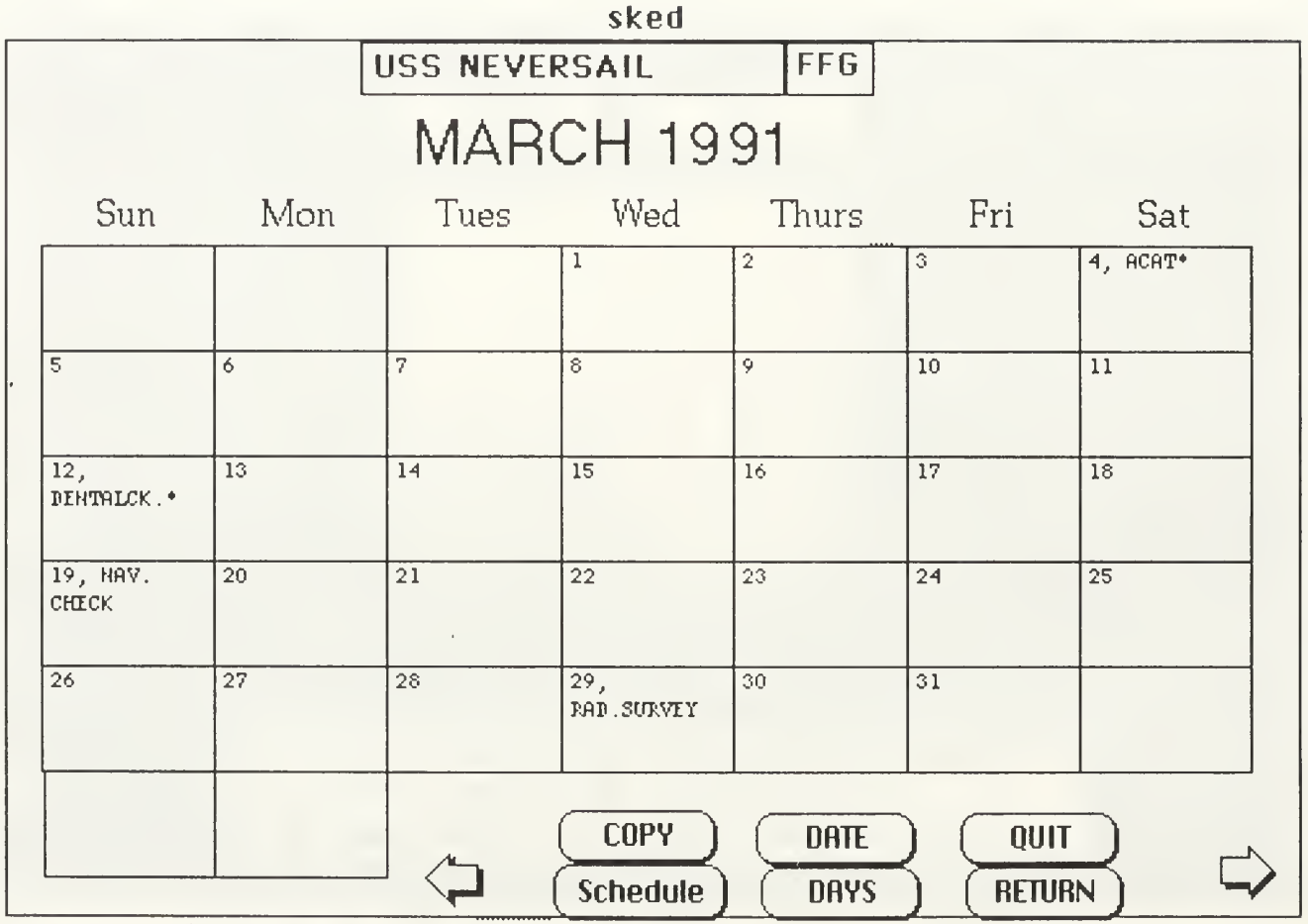

Figure 11
Example Schedule

Argossked. This action will enable the user to see the first card of this stack (Figure 1), which slowly dissolves to the second. The second card contains rectangular buttons which allow the user to perform scheduling actions. Every button in this application which is rectangular has a shadow visual effect. This is done to distinguish these buttons from some of the text fields, which are also rectangular.

The first button which can be used is the BUILD button. When the user clicks the mouse on this button, a box will appear that will require input from the user. This box is called a dialog box, and is used throughout this project. The first dialog box asks the user to input what type of ship will be added to the stack (e.g., FFG, CG, etc.). Next, the user is asked to input the number of a specific type of ship which is currently scheduled. Finally, the user will input the name of the ship. Once this is done, four cards will be added to the end of the stack. The user can then add appropriate information on each card. This function was designed so any number of ships can be added to each stack. (Hypercard has a 512 megabyte stack size limit, however, this is far more than will ever be needed in this project [Ref. 3:p. 129].) The VIEW button is positioned next to the BUILD button and is used to view the information about a ship already scheduled. A dialog box will prompt the user to input the type of ship to be viewed. The use of the BUILD button and the VIEW button cause text fields to

appear. The O.K. button is used to hide these fields. The contents of these fields will remain intact and can be brought into view again by using the BUILD or VIEW buttons.

The next card in each of the ship stacks contains several rectangular buttons (Figure 4). By depressing the EVENT button, the scheduler will see a text field in which inspection prerequisites are listed (Figure 5). The asterisk beside some of the events ensures an event is properly scheduled. The NEXT button is used to compute the next date each event can be scheduled based on the periodicities listed. Finally, the scheduler can view which events were not accomplished by depressing the OVERDUE button.

There are five buttons remaining on this card which perform important functions. The NEXT SKED button sorts the events in the NEXT field in chronological order and then lists these events in the NEXT SKED text field on the third card in the ship stacks (Figure 6). The CURRENT SKED button performs the same function on the events listed in the CURRENT field and then lists those events in the CURRENT SKED text field on the third card. The RESKED button allows the user to reschedule an event. The scheduler inputs the name of the event that must be rescheduled and the application will find that event. The user can then put in a new date.

Moving to the next card (Figure 6), the user will see the current and next schedules listed in chronological order. A certain event can be quickly located by depressing the FIND EVENT button. The VIEW QUARTER button is used to view an entire quarter of either the current or next schedule, in chronological order. The SCHEDULES button located in the lower left hand part of the screen is extremely important. This button will activate the SCHEDULES stack where the events for a ship are actually scheduled.

The computation of fuel consumption is activated by the next card (Figure 7). By depressing the DBSR button, the user will see a list of ship classes and their respective fuel burn rates (Figure 8). Clicking the mouse on the appropriate ship class will cause the appropriate burn rate to appear on this card. To start the process of computing a ship's fuel requirement the mouse is moved to the appropriate employment term and then depressed. A dialog box appears that prompts the scheduler to input the number of days spent underway for this specific event or task. After all terms have been selected, the FUEL REQ'T button will compute the total amount of fuel required in barrels and also the daily quota for days spent underway. Finally, the RESET button located at the bottom of the card must be depressed at the beginning of each new session. This

ensures that text is entered correctly into text fields each time fuel consumption is computed.

The final card in the ship stacks is the Personnel Tempo Reporting Card (Figure 9). The BEGIN button will present a dialog box requesting input for the number of days spent in homeport for a particular quarter. Then, the number of days in the quarter for which the perstempo is being figured is input. The quarterly point total will then be computed and listed. A negative number indicates more days were spent away from homeport than in homeport. A positive number indicates more time spent in homeport vice away from homeport and zero indicates an equal number of days spent in homeport and away from homeport. The CUMULATIVE P/T button calculates cumulative perstempo points and then displays the trend from the previous quarter as improving, declining or steady.

The first card of the SCHEDULES stack contains a card field which holds a list of the ships currently scheduled (Figure 10). The VIEW button allows the user to view the schedule of one of the ships currently scheduled. The QUIT button quits this application and Hypercard. Every other card in the SCHEDULES stack contains six buttons (Figure 11). On each of these, the COPY button is used to create a blank calendar which will be placed at the end of the stack. The most important button on each of these cards is the SCHEDULE button. This button creates the schedule for a

particular ship in the form of a calendar. The DATE button is used to put the correct month and year on the calendar. The DAYS button is used to label the days for a particular month. The QUIT button is used to quit this application and Hypercard. Finally, the RETURN button will move the user back into the ship stacks to the exact card which was originally being used.

In the HELP stack, the RETURN button is the main button which is used (Figure 3). Again, this button will return the user to the exact card which was originally being worked on in one of the ship stacks. This button also utilizes sound. The only other functional buttons in this stack are the arrow icons, which are used to move either forward or backward.

V. BENEFITS OF THE DEVELOPED PROTOTYPE

The Argos Scheduling Module offers many benefits to the users. The most important benefit is that the ship's scheduling process can be improved by automating the process. A small number of examples has been used in this prototype; however, a fully developed system could schedule nearly all of the ship's required activities. There may always be conflicts that cannot be automatically scheduled and perhaps they will be solved manually. Also, as stated earlier, a schedule is advisory. There are several factors involved in ship's scheduling which simply cannot be controlled. For example, a ship may suffer a machinery casualty just prior to getting underway which can delay departure. These situations will always exist, regardless of the scheduling method used. However, an automated scheduling system is beneficial by allowing fewer errors.

One benefit discovered while developing this project is the amount of time that can be saved by automating some of the scheduling process. The elimination of manually searching publications and instructions for information has been achieved in this project. In a fully developed system using a relational database, the elimination of the manual searching will save the user much time. Also, as

demonstrated in the SCHEDULES stack, past schedules can be maintained in an organized manner and reproduced on a moment's notice. The manual process wastes time when operators or schedulers try to locate and understand old schedules.

Another benefit of the Argos Scheduling Module is the standard appearance or format of the final schedules. The manual process allows each scheduler to use many different methods to list activities. As mentioned earlier, timelines drawn on cardboard or paper are some of the more common forms. As changes occur, old schedules are erased or written over. After several changes, these schedules are difficult to read and understand. However, schedules that are produced on a computer in a uniform manner will be much easier to update, read and understand. Also, if one standard method of displaying schedules is approved, the amount of time required to train new personnel will greatly decrease. This is crucial given the constant turnover of personnel.

One of the most important features that had to be considered when designing this project was how the Argos Scheduling Module would work with the other projects already completed. In order for the entire Argos project to be effective, the separate modules must be integrated to offer the greatest benefit to ships. One earlier module completed in the Argos project was designed around the maintenance of

a gas turbine engine. One function of that application was to assist a maintenance man in identifying and ordering a faulty part. Pages from the appropriate technical manuals were available to the user and the application completed most of the paperwork needed to order parts. The Argos Scheduling Module could be used in this case to determine the consequences of a needed part not being locally available. If a few computers were networked together, the maintenance man could look at the current schedule to determine the next underway period. He could then determine if upcoming commitments might be in jeopardy. This can be accomplished by simply placing a button in the maintenance application that will transfer the user to a stack containing a current schedule. Supply personnel could also have access to the same schedules and, based on the availability and location of a needed part, recommend further action, such as a casualty report.

Some of the other projects developed for Argos involve training. In this case, a benefit the Argos Scheduling Module offers if integrated is the ability to determine a ship's upcoming events so training can be scheduled accordingly. For instance, one application allows users to practice the steps required to control flooding due to underwater hull damage. If a ship is scheduled to go to Guantanamo Bay, Cuba, for refresher training, this training

application can be utilized to help personnel train and improve their skills.

There are benefits offered by integrating the scheduling module with the existing application that schedules preventive maintenance checks. Almost every piece of equipment on board U.S. Navy ships has one or more maintenance checks that must be performed within a certain periodicity (e.g., daily, weekly, monthly). Several of these checks require the equipment to be tagged out of service, that is, the equipment cannot be used until the maintenance is completed. On certain equipment this can affect the entire ship. For example, radar, communications, or other electronic equipment can suffer degraded performance because one of the air conditioning units is tagged out for maintenance. This could certainly be the case if a ship were operating in the Caribbean or the Middle East. However, if the preventive maintenance application is integrated with the scheduling application, a schedule would be readily available to all concerned. Thus, this maintenance could be scheduled at a time that would least affect ongoing operations. Alternatively, a maintenance man could check when the next in port period is and perhaps delay maintenance until then. Ideally, current schedules would be reviewed prior to formulating a maintenance plan in order to avoid rescheduling.

The Argos Scheduling Module can be integrated with existing projects to greatly benefit users in performing their daily tasks. As future projects are designed, this scheduling application should reveal many more uses.

VI. CONCLUSIONS

The development of this scheduling prototype in the Hypercard environment demonstrated the usefulness of a fully operational scheduling tool in which schedules can be quickly and accurately produced. These schedules are easy to read and understand and can be changed as operational requirements dictate. The need to search through numerous publications and instructions can be greatly decreased or perhaps eliminated.

An automated scheduling system can also help a scheduler become better at his job. For instance, in Figure 5 those events with prerequisites are marked with an asterisk and the necessary information is provided. This feature should alert a scheduler to these prerequisites at all times. An ability to easily calculate fuel requirements (Figure 7) allows a ship's Operations Officer to analyze his schedule in order to perform the most tasks given a certain amount of fuel.

Although one microcomputer could run all of the applications developed, several computers networked together would be more effective. This way, several members of the crew could have access to scheduling information. As mentioned earlier, schedules are usually posted in well

traveled areas of the ship, but it would be useful if certain people (e.g., each division officer or senior enlisted in each division) could have access to the SCHEDULES stack. This can be accomplished by giving the stack a certain password which can be changed as often as necessary to ensure security (REF. 3:p. 110). In addition, the Operations Officer could prepare a schedule which could then be reviewed by the Commanding Officer (CO) on his computer. The CO could then use several of the functions (fuel requirements, event prerequisites) to verify accuracy. This could be accomplished without a formal meeting which saves everyone time. Also, the CO can perform his review without having to search through all of the applicable manuals.

Although the Argos Scheduling Module has proven successful in several areas, there are features which can be improved. One of these features involves the rescheduling of events. If an event is rescheduled, all of the prerequisites must also be rescheduled. In the prototype, the user must reschedule each prerequisite separately. Relationships between events and prerequisites were not defined or programmed. One improvement would be to design the system so the prerequisites are rescheduled automatically. Another useful feature which could be added would be an automatic calculation of the fuel requirements for a ship given the activities that are scheduled. In the

prototype, this is accomplished using a separate card but the capability to automatically calculate fuel requirements based on a current schedule would be very useful.

As the Navy continues to respond to downsizing efforts and budget reductions as a result of events around the world, ships may spend more time in port. One way to overcome some of the problems associated with this situation is to take advantage of every underway opportunity to maximize training. This will require a schedule which is accurate and easy to modify in order to exploit any and all opportunities. The Argos Scheduling Module can provide these capabilities. It is an application that should help any ship take advantage of available resources in order to maintain a high degree of readiness.

LIST OF REFERENCES

1. Sutton, Frank E., *Hypercard Database Technology As Applied To a Threat Evaluator Reference Tool*, Masters Thesis, Naval Postgraduate School, Monterey, Ca., March 1991.

2. Senn, J. A., *Analysis & Design of Information Systems*, pp. 37-38, McGraw - Hill Publishing Company, 1989.

3. Goodman, D., *The Complete Hypercard 2.0 Handbook*, Bantam Books,Inc.,1990.

4. Giannotti, G. CDR. and Duffy, Kevin LT., *ARGOS: Design and Development of Object-Oriented, Event-driven, Multimedia Database Technology in Support of the Paperless Ship*, Masters Thesis, Naval Postgraduate School, Monterey, CA., December 1988.

5. Long, L. and Long, N., *Computers*, Prentice-Hall Inc., 1990.

6. Paul, J.P., TALK TO ME!, Paul Software Engineering, 513 West "A" Street, Tehachapi, Ca, 93561.

7. Aker, S. Z., and others, *The Mactinosh Bible*, p. 851, Goldstein and Blair, 1991.

8. Commander Naval Surface Group Long Beach, Instruction 3120.1, *Long Range Scheduling Procedures*, pp. 1-17, 20 March 1991.

9. Commander Naval Surface Force United Stataes Pacific Fleet, Instruction 9261.1A, *Fuel Management System*, Enclosures 2, 3 and 4, 18 May 1990.

10. Chief of Naval Operations Instruction 3000.13, *Personnel Tempo Of Operations*, pp. 1-4, 7 February 1990.

APPENDIX

script of card button id 15 = "view"

```
on mouseUp
  -- this button allows a user to view a ship already scheduled
  global counter
  ASK "WHAT TYPE OF SHIP WOULD YOU LIKE TO VIEW"
  if it is ff then
    show card field ff
  else
    if it is dd then
      show card field dd
    else
      if it is ffg then
        show card field ffg
      else
        if it is ddg then
          show card field ddg
        else if it is empty then exit mouseUp
      end if
    end if
  end if
  ASK "PLEASE ENTER THE NAME OF THE SHIP YOU WOULD LIKE TO VIEW"
  hide card field ff
  hide card field dd
  hide card field ffg
  hide card field ddg
  find whole it in card field "ship name"
  if it is empty then exit mouseUp
end mouseUp
```

```
on mouseUp
  ask "WHICH TYPE OF SHIP WOULD YOU LIKE TO ADD?"
  if it is ff then
    show card field ff
  else
    if it is ffg then
      show card field ffg
    else
      if it is dd then
        show card field dd
      else
        if it is ddg then
          show card field ddg
        else if it is empty then exit mouseUp
      end if
    end if
  end if
  unmark all cards -- just to be sure
  global counter -- variable used for writing to ff,ffg,dd,ddg fields
  ask "HOW MANY SHIPS ARE CURRENTLY SCHEDULED?"
  if it is empty then exit mouseUp
  put it into counter
  ASK"PLEASE ENTER THE NAME OF THE SHIP YOU WOULD LIKE TO ADD,FOLLOWED¬
  BY THE SHIP TYPE (e.g. USS ANTRIM,FFG)"
  -- item two is used to indicate which cd field to put it into
  if it is empty then exit mouseUp
  if item 2 of it is "ff" then
    put  item 1 of it into line counter +1 of card field ff
  else
    if item 2 of  it is "ffg" then
      put  item 1 of it into line counter +1 of card field ffg
    else
      if  item 2 of it is "dd" then
        put  item 1 of it into line counter +1 of card field dd
      else
        if  item 2 of it is "ddg" then
          put  item 1 of it into line counter + 1 of  card field ddg
        end if
      end if
    end if
  end if
  lock screen -- the following script is used to add a ship to the stack
  go to card  "EX1"
  doMenu "copy card"
  go to last card of stack
  doMenu "paste card"
  set marked of this card to true -- set to true in order to find later
  go to card "EX2"
  doMenu "copy card"
  go  to last card of stack
  doMenu "paste card"
  go to card "EX3"
  doMenu "copy card"
  go to last card of stack
  doMenu "paste card"
  go to card "EX4"
  doMenu "copy card"
  go to last card of stack
  doMenu "paste card"
  go to first marked card
end mouseUp
```

```
on mouseUp
   -- this script computes the next date of each event based on the
   -- periodicities given. The time periods of day,week and month
   -- are converted to seconds for the computations required.
   put the number of lines of card field "date"into x
   repeat with num = 1 to x
     put  line num  of card field  "date" into datehold
     convert  datehold to seconds
     put line num of card field "periodicity" into line num  of perhold
     if line num of card field "periods" = "weeks"
     then
        multiply line num of perhold  by 604800
        put line num of  perhold into sevendays
        add  datehold to sevendays
        convert sevendays to short date
        put sevendays into line num of card field "ndate"
     else
        if line num of card field "periods" = "months"
       then
          multiply line num of perhold by 2592000
          put line num of perhold into thirtydays
          add  datehold to thirtydays
          convert thirtydays to  short date
          put thirtydays into line num of card field "ndate"
       else
          if line num of card field "periods"= "days"
          then multiply line num of perhold by 86400
          put line num of perhold into  oneday
          add  datehold to oneday
          convert oneday to  short date
          put oneday into line num of card field "ndate"
       end if
     end if
   end repeat
end mouseUp
```

```
on mouseUp
   -- this script calculates the next time for each event based on
   -- the periodicities listed
   put the number of lines of card field "date" into x
   repeat with num = 1 to x
      put line num of card field "date" into line num of getdate
      put line num of card field "ndate" into line num of  getnextdate
      put line num of card field "periodicity" into  line num of interval
      convert line num of getdate to seconds
      convert line num of  getnextdate to seconds
      subtract line num of getdate from line num¬
      of getnextdate
      if line num of card field "periods"= "weeks" then
        multiply line num of interval by 604800
     else
        if line num of card field "periods"="months" then
          multiply line num of interval by 2592000
       else
          if line num of card field "periods" = "days" then
            multiply line num of interval by 86400
         end if
       end if
     end if
     if line num of getnextdate > line num of interval
     then
        put line num of card field "name"¬
        into line num of  card field "tango"
     else put empty into line num of card field "tango"
   end repeat
end mouseUp
```

```
on mouseUp
  -- this button sorts names and dates and then puts them into the
  --  NEXT SKED field on the next card.
  set cursor to watch
  put "I'M WORKING!"
  global holdnextsked
  hide card field "next result" -- used to hold names and dates for sorting
  put the number of lines of card field "name" into x
  repeat with num = 1 to x
    put line num of card field "ndate" into item 1¬
    of line num of card field "next result"
    put line num of card field "name" into item 2¬
    of line num of card field "next result"
  end repeat
  sort lines of card field "next result" dateTime
  put card field "next result" into holdnextsked -- holds names and dates to go to next
  go to next card
  hide message box
  put holdnextsked into card field "nextsked"
end mouseUp
```

```
on mouseUp
  -- this button sorts names and dates and puts them into the
  -- CURRENT SKED field on the next card
  set cursor to watch
  put"I'M WORKING!"
  global holdcurrship
  global nship
  hide card field "result" -- used to hold dates and names for sorting
  set cursor to watch
  put the number of lines of card field "name" into x
  repeat with num = 1 to x
    put line num of card field "date" into item 1 ¬
    of line num of card field "result"
    put line num of card field "name" into item 2 ¬
    of line num of card field "result"
  end repeat
  sort lines of card field "result" dateTime
  put card field "result" into holdcurrsked  -- holds sorted dates and names to go to r
  put card field "ship name" into nship
  go to next card
  hide message box
  put holdcurrsked into card field "skedres"
  put nship into card field "nameship"
end mouseUp
```

```
on mouseUp
  -- this button enables the user to view any quarter of the schedule of
  -- events for the current or next schedule
  set cursor to watch
  put "I'M WORKING. THIS WILL TAKE A FEW SECONDS!"
  hide card field "answer"
  ask "PLEASE ENTER 1, 2, 3, or 4."
  if it is empty then exit mouseUp
  put empty into card field "beta"
  put empty into card field "quarter"
  repeat until it = 1 or it =2 or it =3 or it =4
    ASK " YOU HAVE MADE AN INCORRECT ENTRY. PLEASE ENTER A 1,2,3 OR 4."
  end repeat
  if it is empty then exit mouseUp
  put the number of lines of card field  "skedres" into x
  repeat with num = 1 to x
    put  line num of card field "skedres" into line num of  card field ¬
    "answer"
    convert item 1 of line num of card field "answer" to dateItems
    if it is "1" then
      if  item 2  of line num of card field "answer" < 4
      then
        convert  item 1 to 7 of line num of card field "answer" to short date
        put  line num of  card field "answer" into ¬
        line num of card field "beta"
        put "CURRENT FIRST QUARTER" INTO CARD FIELD "QUARTER"
      end if
    else
      if it is "2" then
        if  item 2 of line num of card field "answer" ≥ 4 ¬
        and   item 2 of line num of card field "answer"¬
        ≤ 6 then
          convert item 1 to 7 of line num of card field "answer" to short date
          put line num of card field "answer" into line num of card ¬
          field "beta"
          put "CURRENT SECOND QUARTER" into card field "quarter"
        end if
      else
        if it is "3" then
          if  item 2 of line num of card field "answer" ≥ 7 ¬
          and  item 2 of line num of card field "answer" ¬
          ≤ 9 then
            convert item 1 to 7 of line num of card field "answer" to short date
            put line num of card field "answer" into line num of¬
            card field "beta"
            put "CURRENT THIRD QUARTER" into card field "quarter"
          end if
        else
          if it is "4" then
            if  item 2 of line num of card field "answer" ≥ 10¬
            and   item 2 of line num of card field "answer"¬
            ≤ 13 then
              convert item 1 to 7 of line num of card field "answer" to short date
              put line num of card field "answer" into line num of ¬
              card field "beta"
              put "CVURRENT FOURTH QUARTER" into card field "quarter"
            end if
          end if
        end if
      end if
    end if
  end repeat
end mouseUp
```

```
on mouseUp
   -- this button performs the calculations necessary to calculate
   -- the fuel required for a ship to perform certain events.
   global counter
   hide cd field "fuel"
   put the number of lines of card field "names" into x
   repeat with num = 1 to x
      put card field "burn" into temp -- hold values to perform multiplication
      multiply temp by item 2 of line num of card field "names"
      multiply temp by line num of card field "underway"
      put temp into line num of card field "fuel"
   end repeat
   put line 1 of card field "fuel" into calcfuel
   repeat with num = 2 to x -- to get totals
     add line num of card field "fuel" to calcfuel
   end repeat
   put calcfuel into cd field "total"
   put line 1 of cd field "underway" into temp1
   repeat with num = 2 to x -- to get totals
     add line num of cd field "underway" to temp1
   end repeat
   divide calcfuel by temp1
   put calcfuel into cd field "totalu/w"
end mouseUp
```

```
on mouseUp
   -- this script calculates the personnel tempo total points
   -- based on input from the user.
   ASK "PLEASE ENTER THE NUMBER OF DAYS THIS SHIP WAS IN HOMEPORT THIS QUARTER"
   put it into alpha
   put it into temp
   ASK "PLEASE ENTER THE NUMBER OF DAYS IN THIS QUARTER"
   put it into beta
   subtract beta from alpha
   put alpha into underway
   add temp to underway
   put underway into card field "total"
   put empty into alpha
   put empty into beta
end mouseUp
```

```
on mouseUp
   -- this computes the personnel tempo points and then computes the
   -- trend from the previous month.
   put card field "ctotal" into temp
   add card field "total" to card field "ctotal"
   if card field "ctotal" > temp then
      put "improving" into card field "trend"
   else
      if card field "ctotal" = temp then
         put "steady" into card field "trend"
      else
         if card field "ctotal" < temp then
            put "declining" into card field "trend"
         end if
      end if
   end if
end mouseUp
```

```
on mouseUp
   global typeofship
   global destination
   repeat with x = 1 to the number of cards
     set the dontSearch of card x to true
   end repeat
   set the dontSearch of this card to false -- so only one is found
   global itemhold
   global counter
   global count
   global fname
   global box
   global holdit
   global temp
   ask "PLEASE ENTER THE NAME OF THE SHIP YOU DESIRE TO SCHEDULE"
   if it is empty then exit mouseup
   put it into holdit
   ask "PLEASE ENTER THE SHIP TYPE (e.g. ffg,cg,ddg,etc.)"
   if it is empty then exit mouseup
   put it into cd field "ship type"
   put cd field "ship type" into typeofship
   if typeofship is "ffg" or typeofship is "ddg" or typeofship is "ff"¬
   or typeofship is "dd" then
      put "Argossked" into destination
   else
      if typeofship is "cg" or typeofship is "cv" then
         put "cruiser/carrier" into destination
      else
         if typeofship is "lsd"  or typeofship is "lst" ¬
         or typeofship is "lha" or typeofship is "lph"or typeofship is "bb" then
           put "amphibs" into destination
         else
            if typeofship is "ao" or typeofship is "ae" or typeofship is "aor"¬
            or typeofship is "aoe" then
               put "replenishment ships" into destination
           end if
         end if
      end if
   end if
   schedule -- this script is in the background
end mouseup
```

```
on schedule
  -- this script prompts the user for information and then based on
  -- the ship type given , goes to the correct stack and retrieves
  -- the scheduling information.This information is then entered into
  -- the correct text field on each calendar.
  global holdit
  global typeofship
  global destination
  global findname
  answer "HAS THIS SHIP ALREADY BEEN SCHEDULED" with "YES" or "NO"
  if it is "no" then
    lock screen
    push card
    go to card hide
    put the number of lines in card field "list" into count
    put holdit into line count + 1 of card field "list"
    pop card
  end if
  put  holdit into cd field "sname"
  set cursor to watch
  put item 1 of cd field "sname" into findname
  put "I'M WORKING"
  lock screen
  go to stack destination -- destination holds the proper ship type
  find whole findname in cd field "ship name"
  go next card
  put cd field "skedres" into comp
  go to stack "sked"
  find whole findname
  put comp into cd field "holder"
  hide msg box
  ask "PLEASE  ENTER THE NUMBER OF THIS MONTH"
  put it into box
  if it is empty then exit schedule
  put "I'M WORKING ! THIS WILL TAKE A FEW MINUTES"
  set cursor to watch
  lock screen -- the following repeat loop schedules events
  put the number of lines of cd field "holder" into counter
  repeat with num = 1 to counter
    convert item 1 of line num of cd field "holder" to dateItems
    if item 2 of line num of cd field "holder" = box then
      put item 3 of line num of cd field "holder" into temp
      find whole temp
      put word 3 of the foundField into fname
      put "," into comma
      put comma && item 8 of line num of cd field "holder"¬
      after last item in cd field fname
    else
      push card
      get line num of cd field "holder"
      go to card hide
      put it into line num of cd field "garbage"
      pop card
    end if
  end repeat
  hide the msg box -- the following repeat loop converts the dateItems
  -- to short dates in card field "holder"
  repeat with x= 1 to the number of lines in cd field "holder"
    convert item 1 to 7 of line x of cd field "holder" to short date
  end repeat
end schedule
```

```
on mouseUp
   -- the following script assigns the proper date to each calendar
   -- based on input from the user
   global day
   global tempyear
   ASK "PLEASE ENTER THE MONTH AND THE YEAR IN THE FOLLOWING MANNER: 1,91"
   put cd field "year1" && item 2 of it into item 1 of tempyear
   put  item 1 of it into item 2 of tempyear
   put item 1 of it into day
   put item 2 of it into cd field "year2"
   if item 1 of it is  "1" then
     put "JANUARY" into cd field "month"
   else
     if item 1 of it is  2 then
       put "FEBRUARY" into cd field "month"
     else
       if item 1 of it is  3 then
         put "MARCH" into cd field "month"
       else
         if item 1 of it is  4 then
           put "APRIL" into cd field "month"
         else
           if item 1 of it is  5 then
             put "MAY" into cd field "month"
           else
             if item 1 of it is  6 then
               put "JUNE" into cd field "month"
             else
               if item 1 of it is  7 then
                 put "JULY" into cd field "month"
               else
                 if item 1 of it is  8 then
                   put "AUGUST" into cd field "month"
                 else
                   if item 1 of it is  9 then
                     put "SEPTEMBER" into cd field "month"
                   else
                     if item 1 of it is  10 then
                       put "OCTOBER" into cd field "month"
                     else
                       if item 1 of it is  11 then
                         put "NOVEMBER" into cd field "month"
                       else
                         if item 1 of it is  12 then
                           put "DECEMBER" into cd field "month"
                         end if
                       end if
                     end if
                   end if
                 end if
               end if
             end if
           end if
         end if
       end if
     end if
   end if
end mouseUp
```

```
on mouseUp
   -- this button numbers the days of each calendar based on input from
   -- the user
   global daycount
   global day
   ask "ON WHAT DAY OF THE WEEK DOES THIS MONTH START?"
   if it is "MONDAY" then
     put 1 into cd field "M1"
     put number of cd field "M1" into whatday
   else
     if it is "TUESDAY" then
       put 1 into cd field "T1"
       put number of cd field "T1" into whatday
     else
       if it is "WEDNESDAY" then
         put 1 into cd field "W1"
         put number of cd field  "W1" into whatday
       else if it is "THURSDAY" then
         put 1 into cd field "TH1"
         put number of cd field "TH1" into whatday
       else
         if it is "FRIDAY" then
           put 1 into cd field "F1"
           put number of cd field  "F1" into whatday
         else
           if it is "SATURDAY" then
             put 1 into cd field "S1"
             put number of cd field  "S1" into whatday
           else
             if it is "SUNDAY" then
               put 1 into cd field "SU1"
               put  number of cd field "SU1" into  whatday
             end if
           end if
         end if
       end if
     end if
   end if
   repeat with x = 1 to 36
     put empty into cd field x
   end repeat

   if day is 1 or day is 3 or day is 5 or day is 7 or day is¬
   8 or day is 10 or day is 12
   then put 31 into daycount
   else
     put 30 into daycount
     if month is 2 then
       if year mod 4 = 0
       then put 29 into daycount
       else put 28 into daycount
     end if
   end if
   repeat with num = 1 to  daycount
     put  num into line 1 of cd field whatday
     add 1 to whatday
   end repeat

end mouseUp
```

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Professor C. Thomas Wu (Code CS/Wq) Computer Science Department Naval Postgraduate School Monterey, California 93942-5000	2
4.	Barry A. Frew Dean of Computers and Information Services (Code 05) Naval Postgraduate School Monterey, California 93940-5000	2
5.	LT. Richard J. Hale 709 Cocklin Street Mechanicsburg, Pennsylvania 17055	2